Лабораторная работа №4: Обработка естественного языка

Цель работы: Познакомиться на практике с методами анализа естественно-языковых текстов в системах логического программирования (Пролог, Mecury), реализовать в соответствии с вариантом задания несложный фрагмент естественного-языкового интерфейса к модельной задаче и протестировать его на ряде примеров.

1. Генеалогическое дерево задано фактами вида

```
parent(alexei,tolia).
parent(alexei,volodia).
parent(tolia,tima).
```

…

Написать программу на Прологе, запросы к которой будут выглядеть следующим образом:

……..

Запросы: ?- answer([volodia, brat , toli, '?'],X).
?- answer([kto, tolin, brat, '?'],X).
?- answer([chei, brat, volodia, '?'],X).

Результаты: X=yes, X=volodia, X=tolia.

---

2. Реализовать разбор предложений английского языка. В предложениях у объекта (подлежащего) могут быть заданы цвет, размер, положение. В результате разбора должны получиться структуры представленные в примере.

……

Запросы: ?- sentence(["The", " big", " book", " is", " under", " the", " table",X).
?- sentence(["The", " red", " book", " is", " on", " the", " table",X).
?- sentence(["The", " little", " pen", " is", " red",X).

Результаты: X= s(location(object(book,size(big)), under(table))),
X= s(location(object(book, color(red)), on(table))),
X= s(object(pen,size(little)), color(red)).

---

3. Реализовать синтаксический анализатор арифметического выражения и вычислить его числовое значение. В выражении допустимы операции +,-,*,/, степень ^. Учитывать приоритеты операций.

……

Запрос: `?- calculate([5, '+', 3, '^',2], X).`

Результат: `X=14`

4. Реализовать синтаксический анализатор арифметического выражения для перевода его в префиксную форму. В выражении допустимы операции +,-,*,/.

……..

Запрос: `?- calculate([5, '+', 2,'*', 3], X).`

Результат: `X=['+',5,'*',2,3]`

5. Реализовать грамматический разбор фраз на ограниченном естественном языке и преобразовать данные фразы в язык исчисления предикатов первого порядка типа:

…….

Запрос: ?- `test([every, man, that, lives, loves, a, woman],Res).`

Результат: Res = all(X, man(X) & lives(X) => exists(Y, woman(Y) & loves(X, Y)))

6. Реализовать синтаксический анализатор логического выражения и вычислить его значение. В выражение допустимы операции отрицания, дизъюнкции, конъюнкции, следования.{~, V, &, =>}. Истинные считаются только термы заданы фактами вида

```
true(river_volga).
true(pupil_vasia).
…
```

……

Запрос*:* ?-calculate(‘(‘,false,‘V’, river_volga,’)’ & true], X).
Результат: X=true.

---

7. Реализовать морфологический разбор глаголов:{выучил, учила, изучили, обучил, ...}.
Корни (уч, ...) ,
приставки (вы, из, об, ...) — содержатся в файле-словаре приставок,
окончания (ил, ила, ило, или) .
Результат должен содержать сведения о роде и числе. Пример:

……

Запрос: ?- an_morf([“и”, “з”, “у”, “ч”, “и”, “л”, “а"], X).
Результат: X=morf(prist(’из’), kor(’уч’), rod(’жен’), chislo(’един’)).

---

8. Реализовать преобразователь активных и пассивных форм типа:
[’Саша’, ’и’, ’Лена’, ’любят’, ‘шоколад’] и
[’шоколад’, ’любим’, ’Сашей’, ’и’, ’Петей’]
в глубинные структуры типа и сравнить полученные глубинные структуры.

……

Запрос: ?- compare([’Саша’, ’и’, ’Лена’, ’любят’, ‘шоколад’] , [’шоколад’, ’любим’, ’Сашей’, ’и’, ’Леной’], Ph1,Ph2, Y).
Результат: Ph1=likes([agent(’Саша’), agent(’Лена’)], object(’шоколад’)),
Ph2=likes([agent(’Саша’), agent(’Лена’)], object(’шоколад’)), Y=yes.

---

9. Реализовать разбор фраз языка (вопросов), выделяя в них неизвестный объекты

……

Запрос: **?-** an_q([“Кто”, “любит”, “шоколад” “?”],X)
?- an_q([“Где”, “лежат”, “деньги” “?”],X)
?- an_q([“Что”, “любит”, “Даша” “?”],X)
Результат: **X**=’любить’(agent(Y), object(’шоколад’)),
X=’лежать’(object(‘деньги’), loc(X)),
X=’любить’(agent(“Даша”), object(Y)).

---

10. Реализовать разбор фраз языка, представляющих собой положительные и отрицательные высказывания. В результате предикат должен выдавать все атомарные глубинные структуры.

……..

Запрос:?- decompose([“Саша” “любит” “игрушки” “,” “но” “не” “любит” “кубики” “и” “мячи”],X);
?- decompose([“Ира” “не” “любит” “стихи” “и” “прозы” “,” “а” “любит” “пьесы”],X)
Результат: **X**=likes(’Саша’, ’игрушки’).
X=not_likes(’Саша’, ’кубики’).
X=not_likes(’Саша’, ’мячи’).

X=not_likes(’Ира’, ’стихи’).
X=not_likes(’Ира’, ’прозы’).
X=likes(’Ира’, ’пьесы’).
